Configuring org.clazzes.login.sql 1.2+

Configuration of sql-login-service 1.2 and higher

The SQL login service may be configured using the OSGi configuration PID `org.clazzes.login.sql` using the configuration values shown in the table below.

Beginning with the version 1.1.0 (released 2013-02-13), all query strings default to the database structure used by the upcoming SDS (SQL Directory Service) bundle. When using another database structure that does not allow some of the queries, it is important set those configuration values to empty strings; deleting them will not help because default values would kick in right away.

Results of list queries (group memberships, group members) are sorted naturally in the Java layer, so there is no need to use ORDER BY clauses. ORDER BY clauses often provoke temporary tables and filesort, which is quite expensive for queries used quite often.

Global configuration directives

Key	Description
`defaultDomain`	Optional. Defaults to an empty string.

Per-Domain configuration directives

Starting with version 1.2.0 sql-login-service supports multiple authentication domains, and therefore began to introduce authentication domain in the names of configuration keys.

The current approach presumes that one database usually provides authentication data for one authentication domain. Support for databases maintaining multiple authentication domains (in one database) might be added in the future (see LOGIN-11), but I do not believe multi-domain-databases even exist outside the LDAP/ADS world.

Version 1.3.2 reflects the changed table names of SDS' first release version 1.0.0, which started to use `SDS_` prefixes for all table names, to make it easier to live in App's databases.

Key	Description
`domain.<domain>.dataSourceName`	Required. Introduced with 1.2.0. Name of the JDBC-Provider's DataSource that provides access to the database containing the authentication data for this authentication domain. Example: `domain.MYAUTHDOMAIN.dataSourceName = MYDATASOURCE`
`domain.<domain>.deactivateUserStatement`	Required non-empty for *deactivateUser* feature. SQL template for a prepared statement to deactivate a user. Default, appropriate for SDS' tables: `UPDATE SDS_USERS SET PASSWORD='{disabled}' WHERE USERID=?`
`domain.<domain>.groupsByUserIdQuery`	Required non-empty for *getGroups* feature. SQL template for a prepared statement to query the group IDs and group names of the groups of which the user specified by a userId is a member. Default, appropriate for SDS' tables: `SELECT g.GROUPID, g.GROUPNAME FROM SDS_GROUPS AS g, SDS_USERS AS u, SDS_GROUPMEMBERSHIPS AS m WHERE u.USERID='?' AND m.USER_ID = u.ID AND g.ID = m.GROUP_ID`
`domain.<domain>.defaultPasswordAlgorithm`	Optional. Defaults to `crypt` Values supported so far: `crypt`, `ssha1`, `plain`. Password fields may contain: • the password encrypted using the default password algorithm, or • a LDAP style algorithm prefix and the password encrypted with the algorithm specified in the prefix. Example: `{PLAIN}badPassword`

`domain.<domain>.setUserPasswordStatement`	Required non-empty for *changePassword* feature. SQL template for a prepared statement to set a new password for the user. Default, appropriate for SDS' tables: `UPDATE SDS_USERS SET PASSWORD=? WHERE USERID=?`
`domain.<domain>.userByUserIdQuery`	SQL template for a prepared statement to query userId, encrypted password, pretty name and e-mail address of a user specified by a userId. If the pretty name is not part of the database, reuse the userId field. If the e-mail address is not part of the database, use a constant like '' or null. Example: `SELECT USERID, PASSWORD, USERNAME, EMAIL FROM SDS_USERS WHERE USERID=?`
`domain.<domain>.usersByGroupIdQuery`	Required non-empty for *getGroupMembers* feature. SQL template for a prepared statement to query the user IDs, user names and e-mail-addresses of the members of the group specified by a groupId. Example: `SELECT u.USERID, u.USERNAME, u.EMAIL FROM SDS_GROUPS AS g, SDS_USERS AS u, SDS_GROUPMEMBERSHIPS AS m WHERE g.GROUPID=? AND m.GROUP_ID = g.ID AND u.ID = m.USER_ID`